Web Crawler Project

Overview:
For the purposes of this project, we define the “Internet” as the test JSON file included and a web crawler as software that requests pages from the “Internet”, parses the content to extract all the links in the page, and visits the links to crawl those pages, to an infinite depth.

Requirements:

1. Use any language and frameworks/libraries you want.
2. Your solution should:
   a. Start with any given **address** value in the list of pages and follow **links** to crawl the remaining pages in the list. For example, if provided “page-01”, your crawler should attempt to visit “page-02” and “page-03”. If provided “page-02”, your crawler should attempt to visit “page-01”
   b. Visit each valid page in a JSON “Internet” exactly once. For example, if more than one page has a link to **page-02**, you should only have to parse **page-02** one time.
   c. Handle all the test JSON “Internet” file provided.
   d. Implement **multi-threading** to visit pages in parallel.
   e. Produce expected output where:
      i. The address of pages that are visited successfully are added to a “Success” collection.
      ii. The address of pages that have already been visited are added to a “Skipped” collection.
      iii. The address of pages that are linked to but do not exist are added to an “Error” collection
3. If you have to make a tradeoff between clean, maintainable code and a complete solution in the time you’re able to spend, we would rather see clean code that could be easily maintained by a team of developers.
4. Share your project on Github or Bitbucket.

Test File:

internet_1.json (provided)

Expected output for internet_1.json

```
Success:
["page-99","page-01","page-04","page-05",
"page-02","page-03","page-08","page-09",
"page-06","page-07"]

Skipped:
["page-01","page-10","page-04","page-05",
"page-02","page-03","page-08","page-09"]

Error:
["page-11","page-00","page-12","page-10","page-13"]
```